Workshop:

Dealing with real-time in real world Hybrid Systems

Pieter van Schaik

Altreonic NV

August 24, 2015

Outline

- Overview of Hybrid Systems
- A Practical Example: Yaw Control
- Summary
- Questions for Discussion

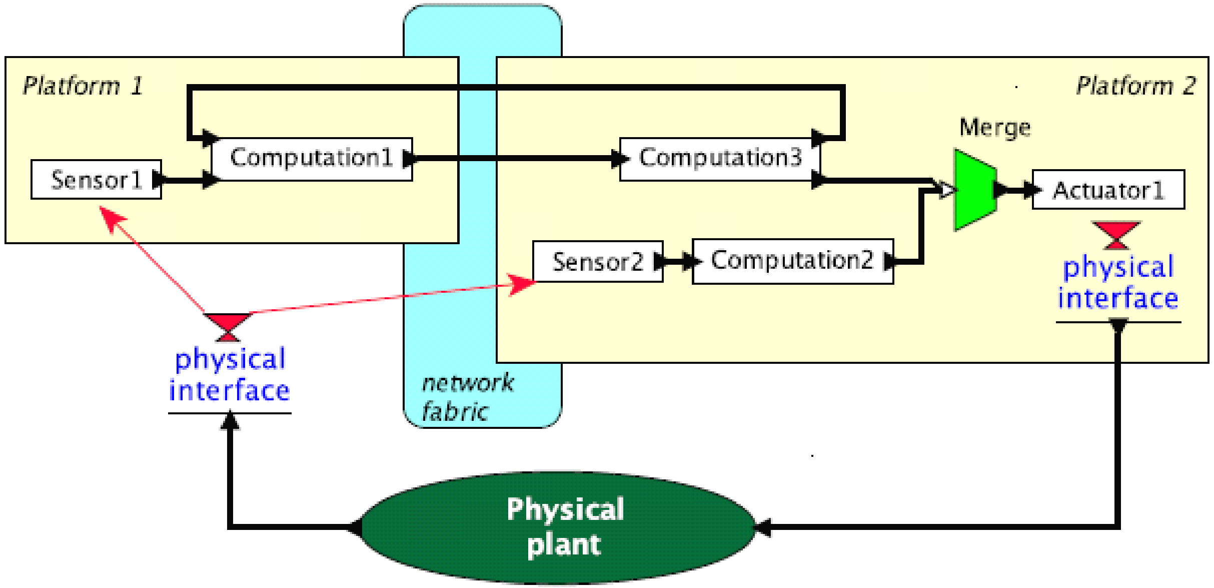
Overview of Hybrid Systems

Abbreviated definition:

"A Hybrid System is a dynamical system with both discrete and continuous state changes"

Simply stated:

A Hybrid System is embedded software controlling a physical process



The Challenge

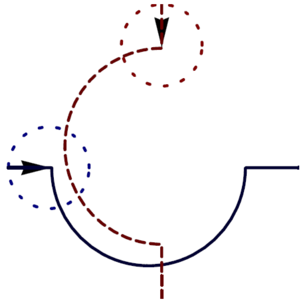
How can we provide people and society with Hybrid Systems that they can trust their lives on?

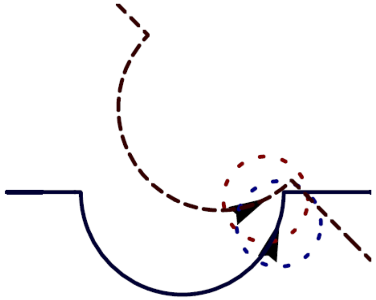
- Methodology to enable compositional certification
  - ➢ Eliminate recertification after integration
- New Formal Modeling Techniques
  - ➢ Conventional models focus on discrete systems

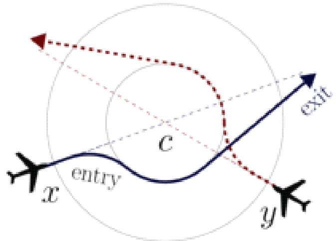
Motivating Examples

Air Traffic Control Systems (ACAS X)

- Differential Dynamic Logic indicated conflicts with actual advisory

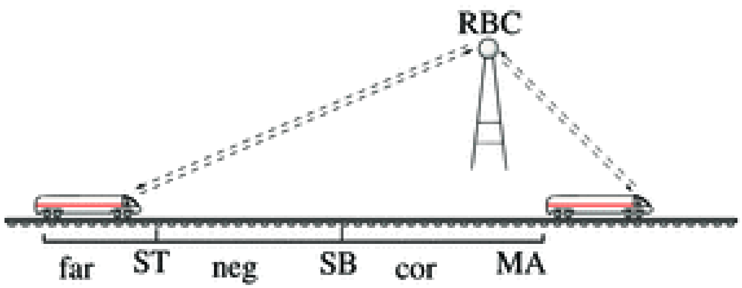






European Train Control System ETCS

- Successful verification of cooperation layer of fully parametric ETCS



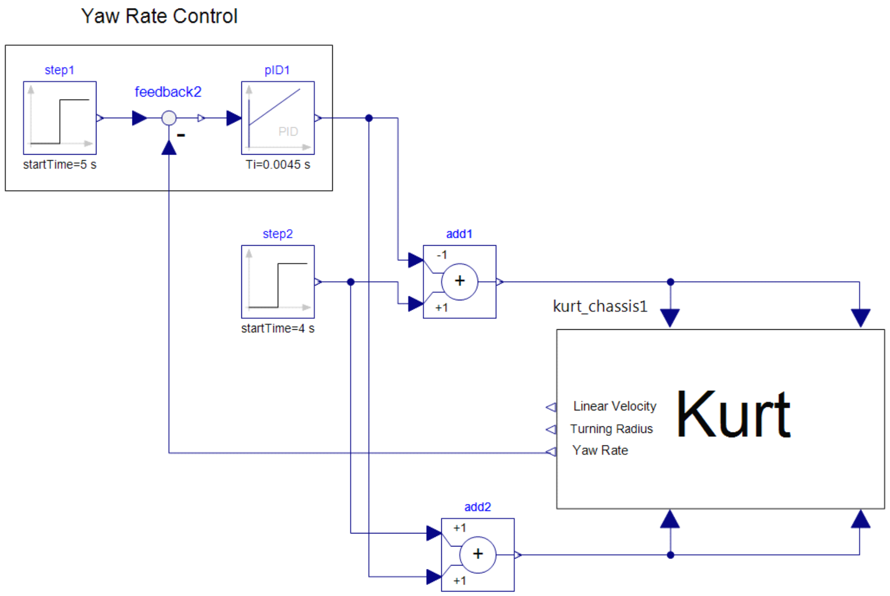
A Practical Example: Yaw Control

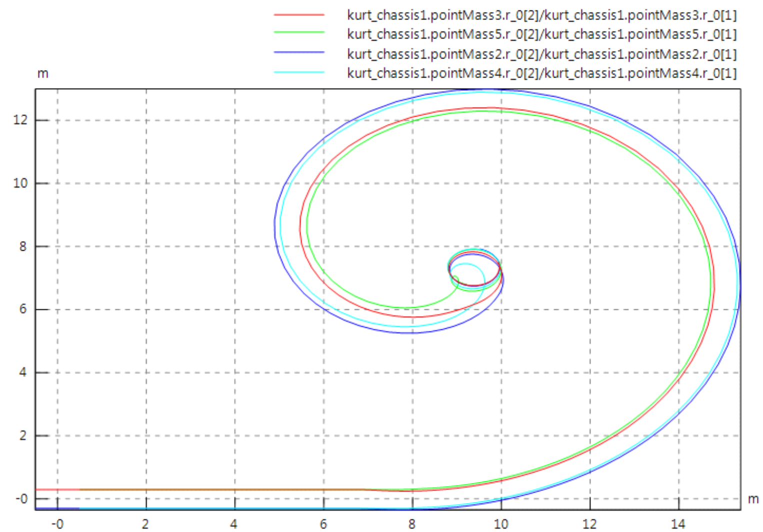
- Goal: Formally model discretization of the KURT skid-steer yaw control
  - ➢ Specific focus on stability of the closed loop system
- Abridged development embedded in Hybrid Event-B formalism

Reference: R. Banach, E.Verhulst, P. van Schaik. Simulation and Formal Modeling of Yaw Control in a Drive-by-Wire Application. *FedCSIS 2015*

Simulations of Yaw Control

- Initial design validation with Modelica simulation
  - ➢ Stability of control strategy
- Simplified PID based control strategy
- PID parameter optimization by practical tuning methods





Modeling Continuous Time Systems

Transfer Function

- Derived from linear time invariant (LTI) differential equation using *Laplace Transform:*

$$F(s) = \int_{0-}^{\infty} f(t) e^{-st} dt$$

$$where\ s = \sigma + j\omega$$

- Transfer function is the ratio of input and output polynomials in *s*, evaluated with zero initial conditions

$$\frac{C(s)}{R(s)} = G(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots b_0}{a_n s^n + a_{n-1} s^{n-1} + \ldots + a_0}$$

- Location of numerator and denominator roots in complex *s-plane* characterise transfer function response

Exponential Stability of LTI Systems

- Exponential stability analysis with transfer function:

$$G(s) = \frac{10(s+4)(s+6)}{(s+1)(s+7)(s+8)(s+10)}$$

- General terms of the output *c(t)* with unit step input:

$$g(t) \equiv A + Be^{-t} + Ce^{-7t} + De^{-8t} + Ee^{-10t}$$

- i.e. any positive real pole causes unstable behaviour

Hybrid Event-B

- Hybrid Event-B - an extension of Event-B
  - All variables are functions of time
  - Mode events and variables - discrete events and variables
  - Pliant events and variables - variables with continuous evolution over time
  - Interfaces allow access to shared variables

```
MACHINE HyEvBMch
TIME t
CLOCK clk
PLIANT x,y
VARIABLES u
INVARIANTS
  x,y,u ∈ ℝ,ℝ,ℕ
EVENTS
  INITIALISATION
    STATUS ordinary
    WHEN
      t = 0
    THEN
      clk,x,y,u := 1,x0,y0,u0
    END
...  ...
```

```
...  ...
  MoEv
    STATUS ordinary
    ANY i?,l,o!
    WHERE
      grd(x,y,u,i?,l,t,clk)
    THEN
      x,y,u,clk,o! :|
        BApred(x,y,u,i?,l,o!,
        t,clk,x′,y′,u′,clk′)
    END
...  ...
```

```
...  ...
  PliEv
    STATUS pliant
    INIT iv(x,y,t,clk)
    WHERE grd(u)
    ANY i?,l,o!
    COMPLY
      BDApred(x,y,u,
        i?,l,o!,t,clk)
    SOLVE
      𝒟x =
        φ(x,y,u,i?,l,o!,t,clk)
      y,o! :=
        E(x,u,i?,l,t,clk)
    END
END
```

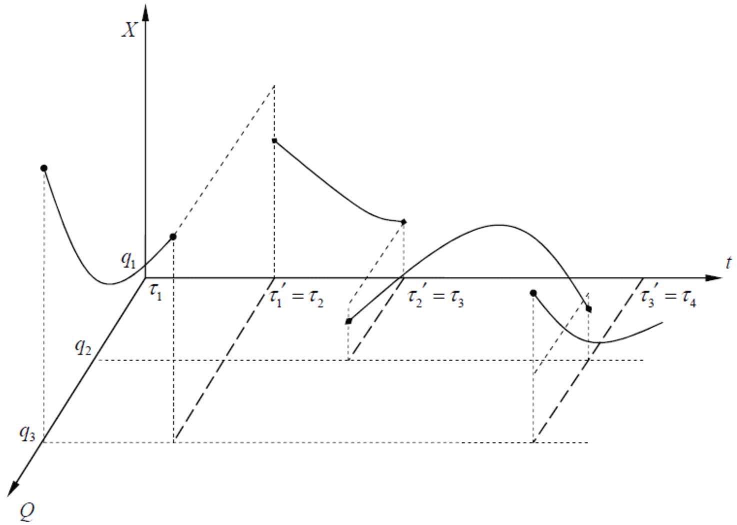
Discrete Event Systems

- Classes of DES models:
  - Untimed DES
    - only concerned with logical behaviour, ex. whether a particular state is reachable
  - Timed DES
    - concerned with both logical behaviour and timing information, ex. whether a particular state is reachable and when it will be reached
- Stability of DES:

  for some set of initial states the system's state is guaranteed to enter a given set and remain there forever

Hybrid Systems

- General Hybrid Dynamical System
  - ➢ dynamic behaviour - differential/difference equations
  - ➢ discrete state space - transition map



- Stability of Hybrid Systems
  - ➢ dynamic behaviour stability - exponential stability
  - ➢ properties of the transition map

Formal Modeling Yaw Control

- KURT yaw rate mathematical model:

$$\frac{d}{dt} yrm(t) = C_k stc(t)$$

- PID controller mathematical model:

$$stc(t) = K_p[yre(t) + \frac{1}{T_I}\int_0^t yre(s)ds + T_D \frac{d}{dt} yre(t)]$$

- Substituting *yre(t) = YRR - yrm(t)* results in:

$$(T_D + \frac{1}{C_k K_P})\frac{d^2}{dt^2} stc(t) + \frac{d}{dt} stc(t) + \frac{1}{T_I} stc(t) = 0$$

- Exponential stability requires that:

$$T_I > 0 \text{ and } T_D + \frac{1}{C_k K_P} > 0$$

Continuous Time HEB Model

- Equivalent Hybrid Event-B system:

```
PROJECT  Kurt_Prj
INTERACES
  YawCtrl_IF
MACHINES
  KurtUser_Mch
  Kurt_Mch
  YawCtrl_Mch
END
```

```
INTERFACE  YawCtrl_IF
SEES  Kurt_Ctx
TIME  t
PLIANT
  yrr, yrm, stc,
  yreP, yreI, yreD,
  thr, tal, tar
INVARIANTS
  yrr, yrm, stc ∈ ℝ, ℝ, ℝ
  yreD, yreP, yreI ∈ ℝ, ℝ, ℝ
  thr, tal, tar ∈ ℝ, ℝ, ℝ
INITIALISATION
  WHEN
    t = 0
  THEN
    yrr, yrm, stc  :=  0, 0, 0
    yreP, yreI, yreD  :=  0, 0, 0
    thr, tal, tar  :=  0, 0, 0
  END
END
```

```
CONTEXT  Kurt_Ctx
  …   …
AXIOMS
  …   …
END
```

```
MACHINE  KurtUser_Mch
CONNECTS  YawCtrl_IF
EVENTS
  SteerKurt
    STATUS  pliant
    BEGIN
      thr(t)  :=  Θ(4 − t)
      yrr(t)  :=  Θ(t − 5)
    END
END
```

```
MACHINE  Kurt_Mch
CONNECTS  YawCtrl_IF
EVENTS
  KurtBehaves
    STATUS  pliant
    SOLVE
      𝒟yrm(t)  :=  C_K stc(t)
    END
END
```

```
MACHINE  YawCtrl_Mch
CONNECTS  YawCtrl_IF
EVENTS
  YawControl
    STATUS  pliant
    SOLVE
      yreP(t)  :=  yrr(t) − yrm(t)
      yreD(t)  :=  𝒟yreP(t)
      𝒟yreI(t)  =  yreP(t)
      stc(t)  :=
        K_P[yreP(t) + yreI(t)/T_I + T_D yreD(t)]
      tal(t)  :=  thr(t) − stc(t)
      tar(t)  :=  thr(t) + stc(t)
    END
END
```

General Model of Yaw Control

Addressing more arbitrary steering episodes requires solving for:

$$\frac{d}{dt}\mathbf{stc}(t) = \mathbf{Astc}(t) + \mathbf{b}(t)$$

where **A** is constant, **stc***(t)* depends on *stc(t)* and *stc'(t)*, **b***(t)* is dependent on the inhomogeneous term:

$$inh(t) = \frac{1}{C_K}\left(T_D \frac{d^3}{dt^3} yrr(t) + \frac{d^2}{dt^2} yrr(t) + \frac{1}{T_I}\frac{d}{dt} yrr(t)\right)$$

Discretizing Yaw Control

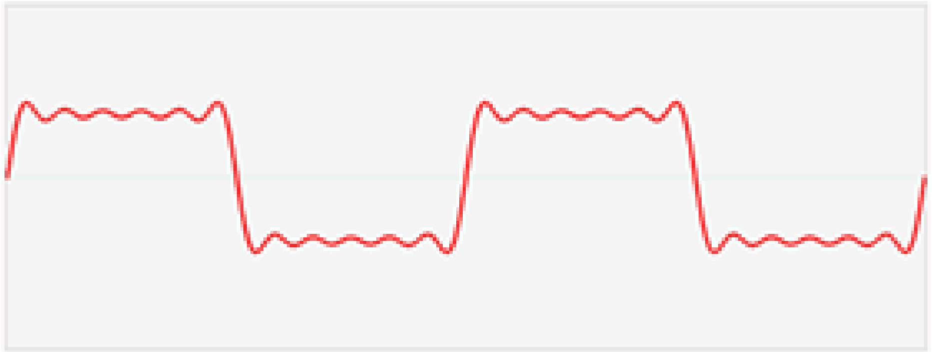
Discretizing Hybrid Event-B Yaw Control

- Implementation on a discrete computing platform requires sampling
- Strategy of viewing discretizing as a refinement poses difficulties:
  - formal standpoint is sampling impoverishes the continuous model
  - degrades information available for consistency proof
- Argument for HEB approach:
  - stability of the discretized system ensures that the system can be steered to a desired regime

Sampled Data Systems

- Sampling frequency must be related to characteristics of function being sampled
  - Sampling frequency too low -> loss of important information
  - Sampling frequency too high -> unnecessarily cost/complexity
- Important to understand the effects of sampling

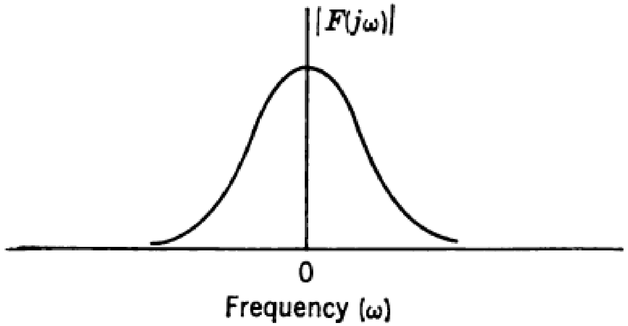
Single Bandwidth Illustration

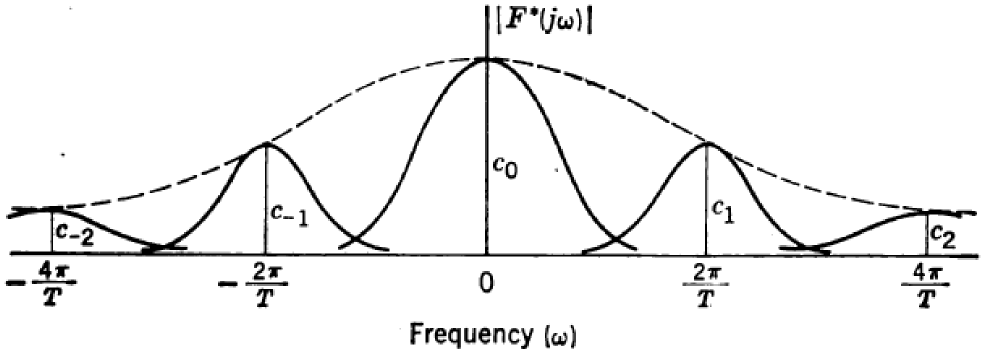


https://en.wikipedia.org/wiki/File:Fourier_series_and_transform.gif

Effects of Sampling

Pictorial representation of the effect of sampling:



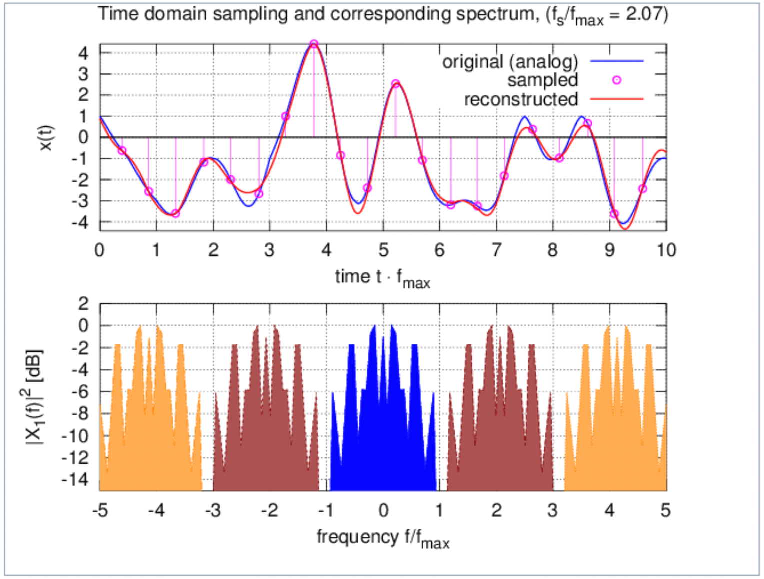


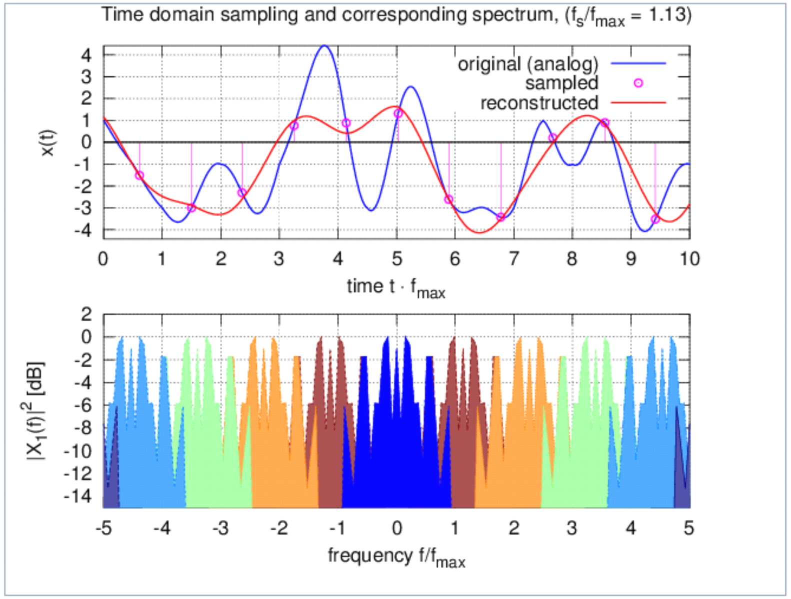
- The central signal spectrum can be recovered by low pass filtering (anti-aliasing filter)
- Shannon-Nyquist theorem limits sampling interval:

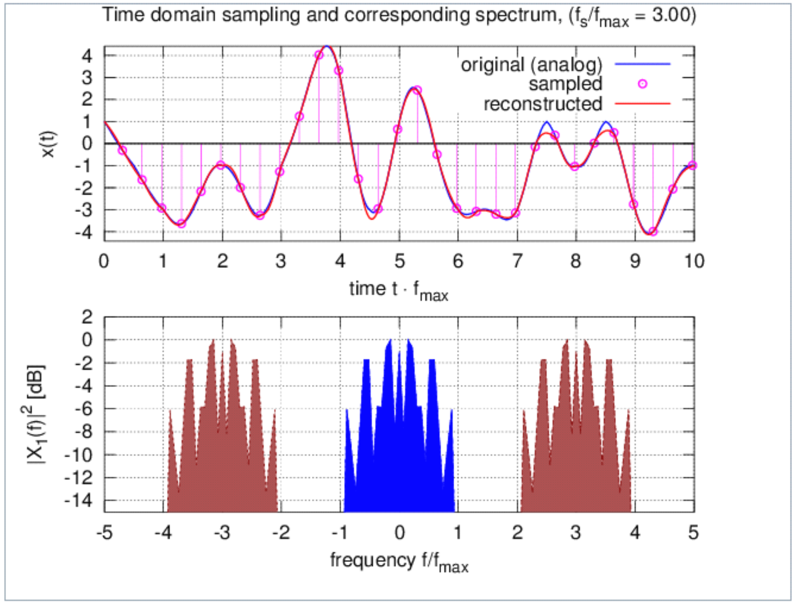
  For band limited signals:

$$T_{s_{\max}} = \frac{\pi}{W}$$

Sampling Effect Illustration







Stability of Sampled Data Systems

- Sampling period affects stability:

  Example: Consider the following SDS transfer function:

$$T(z) = \frac{10(1 - e^{-T})}{z - (11e^{-T} - 10)}$$

For $T > 0.2$ the resulting transfer function is unstable

Discretized HEB Yaw Control

Resulting discretized Hybrid Event-B model:

PROJECT *KurtD_Prj*
REFINES –??– *Kurt_Prj*
INTERFACES
  YawCtrlD_IF
MACHINES
  KurtUserD_Mch
  KurtD_Mch
  YawCtrlD_Mch
END

INTERFACE *YawCtrlD_IF*
REFINES –??– *YawCtrl_IF*
SEES *KurtD_Ctx*
TIME t
PLIANT
  $yrr_D, yrm_D,$
  $stc_D, stc_D^{pr},$
  $yreP_D, yreP_D^{pr},$
  $yreI_D, yreD_D,$
  thr_D, tal_D, tar_D
INVARIANTS
  $yrr_D, yrm_D \in \mathbb{R}, \mathbb{R}$
  $stc_D, stc_D^{pr} \in \mathbb{R}, \mathbb{R}$
  $yreP_D, yreP_D^{pr} \in \mathbb{R}, \mathbb{R}$
  $yreI_D, yreD_D \in \mathbb{R}, \mathbb{R}$
  $thr_D, tal_D, tar_D \in \mathbb{R}, \mathbb{R}, \mathbb{R}$
  $thr_D = thr$
  $yrr_D = yrr$
  $|yrm_D - yrm| < B_{yrm}$
  $|stc_D - stc| < B_{stc}$
  $|stc_D^{pr} - stc| < B_{stc}$
  $|yreP_D - yreP| < B_{yreP}$
  $|yreP_D^{pr} - yreP| < B_{yreP}$
  $|yreI_D - yreI| < B_{yreI}$
  $|yreD_D - yreD| < B_{yreD}$
  $|tal_D - tal| < B_{tal}$
  $|tar_D - tar| < B_{tar}$
… …

… …
  INITIALISATION
    WHEN
      $t = 0$
    THEN
      $yrr_D, yrm_D := 0, 0$
      $stc_D, stc_D^{pr} := 0, 0$
      $yreP_D, yreP_D^{pr} := 0, 0$
      $yreI_D, yreD_D := 0, 0$
      $thr_D, tal_D, tar_D := 0, 0, 0$
    END
  END

CONTEXT *KurtD_Ctx*
EXTENDS *Kurt_Ctx*
  … …
AXIOMS
  $NT = 1$
  … …
END

MACHINE *KurtUserD_Mch*
REFINES *KurtUser_Mch*
CONNECTS *YawCtrlD_IF*
EVENTS
  SteerKurt
    REFINES *SteerKurt*
    STATUS pliant
    BEGIN
      $thr_D(t) := \Theta(4 - t)$
      $yrr_D(t) := \Theta(t - 5)$
    END
END

MACHINE *KurtD_Mch*
REFINES –??– *Kurt_Mch*
CONNECTS *YawCtrlD_IF*
EVENTS
  KurtBehavesPli
    REFINES *KurtBehaves*
    STATUS pliant
    COMPLY skip
    END
  KurtBehavesMo
    STATUS ordinary
    WHEN $(\exists n \in \mathbb{N} \bullet t = nT)$
      $yrm_D := yrm_D + C_K T stc_D$
    END
END

MACHINE *YawCtrlD_Mch*
REFINES –??– *YawCtrl_Mch*
CONNECTS *YawCtrlD_IF*
EVENTS
  YawControlPli
    REFINES *YawControl*
    STATUS pliant
    COMPLY skip
    END
  YawControlMo
    STATUS ordinary
    WHEN $(\exists n \in \mathbb{N} \bullet t = nT)$
      $yreP_D := yrr_D - yrm_D$
      $yreP_D^{pr} := yreP_D$
      $yreI_D := yreI_D + TyreP_D$
      $yreD_D := (yreP_D - yreP_D^{pr})/T$
      $stc_D :=$ "$K_P[yreP_D + yreI_D/T_I + T_D yreD_D]$"
      $stc_D^{pr} := stc_D$
      $tal_D := thr_D - stc_D$
      $tar_D := thr_D + stc_D$
    END
END

A Practical Example: Yaw Control

Discretized Stability Analysis

- A similar approach to analogue counter part resulted in:

$$stc_{D,k+3} - 2stc_{D,k+2} + stc_{D,k+1} = -C_K K_P[T_D(stc_{D,k+2} - 2stc_{D,k+1} + stc_{D,k})$$
$$+ T(stc_{D,k+2} - stc_{D,k+1}) + T^2 stc_{D,k+2} / T_I]$$

- Requires solving for:

$$W^3 + C_k K_P[T^2 / T_I + T + T_D - 2 / C_k K_P]W^2 + C_k K_P[1 / C_k K_P - 2T_D - T]W$$
$$+ C_k K_P T_D = 0$$

- For stability, eventually deduce:

$$1 > C_k K_P T_D$$

Summary

- Viewing discretization as an instance of refinement is demanding
- Many simplifications required to render calculations tractable
    - mathematical insight and domain knowledge required
- Closer cooperation needed between frequency domain and state space approaches

Questions for Discussion

- Can sampling theory be applied to reconcile continuous and discrete views in a way that is acceptable to formal techniques?
- Can supporting tools make hybrid system formal methods more accessible to engineers?